



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/826,973	04/16/2004	Gregory E. Niles	18602-08906 (P3331US1)	1031
61520	7590	02/19/2009		
APPLE/FENWICK SILICON VALLEY CENTER 801 CALIFORNIA STREET MOUNTAIN VIEW, CA 94041			EXAMINER RIEKO, JASON MICHAEL	
			ART UNIT	PAPER NUMBER
			2628	
			MAIL DATE	DELIVERY MODE
			02/19/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/826,973

Applicant(s)

NILES ET AL.

Examiner

Jason M. Repko

Art Unit

2628

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 86 and 112-121 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 86, 112-121 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. ____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SG/US)
Paper No(s)/Mail Date ____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date ____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: ____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. **Claims 112, 115, 116, 118, 120, and 121 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,714,201 to Grinstein et al.**

3. Regarding claim 112, Grinstein et al. disclose “in a computer-implemented animation system, a method for animating an object, the method comprising:

- a. receiving an input (*receiving the instructions written in C++ that invoke the OpenMotion API as described in section 6.2 at column 15*) specifying an Attracted To behavior (*motion behaviors are described in section 6.2.6, where it is disclosed “A Behavior is an action that changes a motion’s parameters” in lines 12-14 of column 29; the behavior of ball1 and ball2 as described in column 37 in section 6.2.8.2,*

- i. the Attracted To behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object (*The position parameter of ball 1 and ball2 described in column 37 are changed over according to the trajectory and the velocity parameter as described in section 6.2.5.1 at column 25 and Table 10: “dv/dt, time derivative of position.” Referring to lines 13-33 of column 37, the*

trajectory, and thus position, of ball1 is changed based on the "reflect" behavior action, described in Table 23 in column 35, according to the myBox object boundary position without affecting the position of myBox object.);

- b. animating the object by changing the value of the position parameter of the object over time according to the Attracted To behavior (*Figure 3 shows the communication between the motion 122 and the renderer 106 to create an image for display 124; See also lines 46-53 of column 75: "...generating an animated view of the given model in which the given model is rendered at each of a succession of time values with individual ones of the model's nodes being shown in each successive rendering as having a position and orientation determined as a function of the value for the rendering's corresponding time value of the position and orientation values defined by the node's associated motion..."*; section 6.2.6 describes Behaviors which changing the value of a parameter, for example position, over time); and
- c. outputting the animated object (*the end result of applying motions and behaviors is an animated image of an object as shown in Fig. 3 and described in lines 17-20 of column 53: "Since Mojo is a real-time motion editor, the model of the running man is shown moving according to a set of motions that have been applied to the individual nodes of its hierarchical model."*);
- d. wherein the Attracted To behavior can be modified using
 - ii. a falloff rate parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines a rate of acceleration with which the object move towards the second object (*ball1.behavior can be*

modified using the accelerationControl, described in Table 14 of column 29, to set the acceleration parameter disclosed in Table 10 of column 26, which determines the rate of acceleration of the ball1 object toward the myBox object.);

iii. an influence parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior (ball1.behavior can be modified using boundary expressions shown in Table 20 of column 33 which establish, for example, a rectangular box boundary. The example in lines 13-33 of column 37, modifies the behavior based on the proximity to the area defined by the myBox object boundary.);

iv. a falloff type parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines whether a distance defined by the influence parameter falls off linearly or exponentially (ball1.behavior can be modified based a condition expression as described lines 57-60 of column 30, the condition expression relying on the distance boundary attribute, described in Table 19 of column 33. The action performed in response to the condition can be the velocityControl, described in Table 14 of column 29, to set the velocity parameter, disclosed in Table 10 of column 26, exponentially or linearly.);

v. a strength parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a speed at which the object moves towards the second object (ball1.behavior can be modified using the

velocityControl, described in Table 14 of column 29, to set the velocity parameter disclosed in Table 10 of column 26, which determines the rate of speed of the ball1 object toward the myBox object.); and

vi. *a drag parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines whether the object overshoots the second object (the orientation quaternion from Table 10 can be used as a condition, described lines 57-60 of column 30, to change a velocity parameter in a Behavior object).*

4. Regarding claim 115, Grinstein et al. discloses "in a computer-implemented animation system, a method for animating an object, the method comprising:

e. *receiving an input (receiving the instructions written in C++ that invoke the OpenMotion API as described in section 6.2 at column 15) specifying an Orbit Around behavior (motion behaviors are described in section 6.2.6, where it is disclosed "A Behavior is an action that changes a motion's parameters" in lines 12-14 of column 29; the behavior of ball1 and ball2 as described in column 37 in section 6.2.8.2),*

f. *the Orbit Around behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object (The position parameter of ball1 and ball2 described in column 37 are changed over according to the trajectory and the velocity parameter as described in section 6.2.5.1 at column 25 and Table 10: "dv/dt, time derivative of position." Referring to lines 13-33 of column 37, the trajectory, and thus position, of ball1 is changed based on the "reflect" behavior action, described in Table*

23 in column 35, according to the myBox object boundary position without affecting the position of myBox object. Likewise, ball1 behavior can be modified to move around an object around the myBox boundary in the "Orbital Motion" disclosed in 6.3.3.8 of column 43, using the controllers described in Table 14 of column 29 to set the parameters.);

g. animating the object by changing the value of the position parameter of the object over time according to the Orbit Around behavior (lines 46-53 of column 75:

"...generating an animated view of the given model in which the given model is rendered at each of a succession of time values with individual ones of the model's nodes being shown in each successive rendering as having a position and orientation determined as a function of the value for the rendering's corresponding time value of the position and orientation values defined by the node's associated motion..."; section 6.2.6 describes Behaviors which changing the value of a parameter, for example position, over time); and

h. outputting the animated object (the end result of applying motions and behaviors is an animated image of an object as shown in Fig. 3 and described in lines 17-20 of column 53: "Since Mojo is a real-time motion editor, the model of the running man is shown moving according to a set of motions that have been applied to the individual nodes of its hierarchical model.");

i. wherein the Orbit Around behavior can be configured regarding:

vii. a falloff rate parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a rate of acceleration

which the object moves around the second object (*ball1.behavior can be modified to move around an object around the myBox boundary in the "Orbital Motion" disclosed in 6.3.3.8 of column 43, using the controllers described in Table 14 of column 29 to set the parameters*);

viii. an influence parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines an area of influence determining whether the object is affected by the Orbit Around behavior (*ball1.behavior can be modified using boundary expressions shown in Table 20 of column 33 which establish, for example, a rectangular box boundary. The example in lines 13-33 of column 37, modifies the behavior based on the proximity to the area defined by the myBox object boundary.*);

ix. a falloff type parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines whether a distance defined by the influence parameter falls of linearly or exponentially (*ball1.behavior can be modified based a condition expression as described lines 57-60 of column 30, the condition expression relying on the distance boundary attribute, described in Table 19 of column 33. The action performed in response to the condition can be the velocityControl, described in Table 14 of column 29, to set the velocity parameter, disclosed in Table 10 of column 26, exponentially or linearly.*); and

x. a strength parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines a speed at which the object moves

around the second object (*ball1.behavior can be modified using the velocityControl, described in Table 14 of column 29, to set the velocity parameter disclosed in Table 10 of column 26, which determines the rate of speed of the ball1 object toward the myBox object.*).

5. Regarding claim 116, Grinstein et al. discloses "in a computer implemented animation system, a method for animating an object, the method comprising:

j. receiving an input specifying a Random Motion behavior (*receiving the instructions written in C++ that invoke the OpenMotion API as described in section 6.2 at column 15*) specifying an Attracted To behavior (*motion behaviors are described in section 6.2.6, where it is disclosed "A Behavior is an action that changes a motion's parameters" in lines 12-14 of column 29*),

xi. the Random Motion behavior indicating how to change a value of a position parameter of the object over time based on a random motion path (*the random wandering behavior for the ball, where the velocity direction is randomized as described in section 6.2.8.5 in lines 38-52 of column 38: " BehaviorVar wander=(velocityControl(randomDir(sinTime())); ... Motion Ball; Ball.behavior (wander,");*

k. animating the object by changing the value of the position parameter of the object over time according to the Random Motion behavior (*lines 46-53 of column 75: "...generating an animated view of the given model in which the given model is rendered at each of a succession of time values with individual ones of the model's nodes being shown in each successive rendering as having a position and orientation determined as a*

function of the value for the rendering's corresponding time value of the position and orientation values defined by the node's associated motion..."; section 6.2.6 describes Behaviors which changing the value of a parameter, for example position, over time); and

l. *outputting the animated object (the end result of applying motions and behaviors is an animated image of an object as shown in Fig. 3 and described in lines 17-20 of column 53: "Since Mojo is a real-time motion editor, the model of the running man is shown moving according to a set of motions that have been applied to the individual nodes of its hierarchical model.");*

m. *wherein the Random Motion behavior can be configured regarding:*

xii. *an amount parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a length of the motion path (the parameter "when (isInside(goal, Ball)).perform(stop)" terminates the motion path, and thus, determines the length of the path);*

xiii. *a frequency parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a crookedness of the motion path (the wander parameter described in lines 38-40 of column 38 determines the direction as a dynamic function of "simTime", and thus, determines the characteristics such as crookedness of the motion path);*

xiv. *a noisiness parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a level of jaggedness along the motion path (the wander parameter described in lines 38-40 of column*

38 determines the direction as a dynamic function of "simTime", and thus, determines the characteristics such as level of jaggedness of the motion path); and
xv. *a drag parameter (a parameter given to a Behavior object as described in lines 21-28 of column 29), which determines a speed at which the object moves along the motion path (the wander parameter not only gives a direction but also determines that the "ball will have a constant speed of 1 unit per second" as described in lines 37-38 of column 38).*

6. Claims 118, 120, and 121 recite limitations similar in scope to the limitations of claim 112, 115, and 116, but as a method on a computer program product. As shown in the rejection, the Grinstein et al. disclose the limitations of claims 112, 115, and 116. Additionally, Grinstein et al. disclose a computer program product for animating an object, the computer program product comprising a computer-readable storage medium containing computer program code in lines 51-56 of column 6: "It should be understood that the invention is meant to include apparatus, methods, computer programming recorded on computer readable media, and propagated signals capable of providing functionality of the type recited in each claim, even if there currently are not claim covering each of these different class of inventions below." Thus, the computer readable medium recited in claims 118, 120, and 121 are met by Grinstein et al. according to the mapping presented in the rejection of claims 112, 115, and 116 because the computer readable medium stores the method disclosed in claim 112, 115, and 116.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. **Claims 86 and 117 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,714,201 to Grinstein et al. in view of U.S. Patent Application No. 2002/0003540 to Unuma et al.**

8. Regarding claim 86, Grinstein et al. disclose “in a computer implemented animation system, a method for animating an object, the method comprising:

- n. receiving an input (*receiving the instructions written in C++ that invoke the OpenMotion API as described in section 6.2 at column 15*) specifying an Align to Motion

behavior (*motion behaviors are described in section 6.2.6, where it is disclosed "A Behavior is an action that changes a motion's parameters" in lines 12-14 of column 29*),

xvi. the Align to Motion behavior indicating how to change a value of a rotation parameter of the object (*Rotation parameters spin and revolve are described in Table 10 in col. 26. These parameters can be changed using actions such as those described in Tables 14-16. For example, spinControl and revolveControl change the value of rotation parameters.*) over time (*Table 18, column 32 describes temporal Behavioral attributes and predicates*) based on a motion path of the object (*6.2.6.4 of columns 30-31 disclose triggers set to affect the behavior based on the motion path*);

o. animating the object by changing the value of the rotation parameter of the object over time according to the Align to Motion behavior (*lines 46-53 of column 75:*

"...generating an animated view of the given model in which the given model is rendered at each of a succession of time values with individual ones of the model's nodes being shown in each successive rendering as having a position and orientation determined as a function of the value for the rendering's corresponding time value of the position and orientation values defined by the node's associated motion..."; section 6.2.6 describes Behaviors which changing the value of a parameter, for example position, over time);
and

p. outputting the animated object (*the end result of applying motions and behaviors is an animated image of an object as shown in Fig. 3 and described in lines 17-20 of column 53: "Since Mojo is a real-time motion editor, the model of the running man is*

shown moving according to a set of motions that have been applied to the individual nodes of its hierarchical model."");

q. wherein the Align to Motion behavior (*motion behaviors described in section 6.2.6*) can be configured regarding

xvii. a spring tension parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines how quickly the objects rotation changes based on a change in the objects motion path (*An object's motion path, e.g. position vector in Table 10, can be used as a condition as described lines 57-60 of column 30: "The when (<condition>) clause contains the trigger condition; the .perform(<behavior>) clause specifies the new state when and if that trigger fires."*. A spinControl action of the behavior from Table 14 can determine how quickly the objects rotation changes by controlling the magnitude of the spin vector shown in Table 10.);

xviii. an axis parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines whether the object's rotation is based on an X value of the object's position or a Y value of the object's position (*the X or Y value of an object's position vector from Table 10 can be used as a condition, described lines 57-60 of column 30, to change the spin parameter using a spinControl action of the behavior from Table 14*); and

xix. a drag parameter (*a parameter given to a Behavior object as described in lines 21-28 of column 29*), which determines whether or not the object's change in rotation overshoots a new direction of the object (*the orientation quaternion from*

Table 10 can be used as a condition, described lines 57-60 of column 30, to change the state of the behavior).

9. Grinstein et al. does not expressly disclose “the Align to Motion behavior indicating how to change a value of a rotation parameter of the object over time based on a motion path of the object such that the rotation parameter is not changed if the motion path is straight.”

10. Regarding claim 86, Unuma et al. disclose “an Align to Motion behavior, which changes a rotation of the object based on a motion path of the object such that the rotation is not changed if the motion path is straight” (*paragraph [0131]: “In this graphic image, the person stands on the ground in a vertical direction i.e. along the z-axis. First, the transit point specifying unit 81 designates a transit point 401 of the person in FIG. 16. The moving direction controller 82 then rotates the object 1 about the y axis so that the front side thereof faces to the transit point 401.”*). For example, if the object’s 1 path is straight, e.g. the transit point 401 is placed in the direction the object 1 is facing, then there is no need to rotate to face the transit point 401. Thus, “the rotation is not changed if the motion path is straight.” It should be noted that Unuma et al. reads on the interpretation that “object’s rotation” is the object’s rotational orientation or the object’s rotational motion.

11. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to incorporate “changes a rotation of the object based on a motion path of the object such that the rotation is not changed if the motion path is straight” as taught by Unuma et al. in the system disclosed by Grinstein et al. by creating a motion with transitions, described in section 6.2.6.4 at column 30, conditional upon the orientation parameter of an object, described in Table 10 at column 26. The motivation for doing so would have been to realistically model human

motion, given humans face the direction they are walking under ordinary circumstances.

Therefore, it would have been obvious to combine Grinstein et al. with Unum et al. to obtain the invention specified in claim 86.

12. Claims 117 recites limitations similar in scope to the limitations of claim 86, but as a method on a computer program product. As shown in the rejection, the combination of Grinstein et al. with Unum et al. disclose the limitations of claim 86. Additionally, Grinstein et al. discloses a computer program product for animating an object, the computer program product comprising a computer-readable storage medium containing computer program code in lines 51-56 of column 6: "It should be understood that the invention is meant to include apparatus, methods, computer programming recorded on computer readable media, and propagated signals capable of providing functionality of the type recited in each claim, even if there currently are not claim covering each of these different class of inventions below." Thus, the computer readable medium recited in claim 117 is met by the combination of Grinstein et al. with Unum et al. according to the mapping presented in the rejection of claim 86 because the computer readable medium stores the method disclosed in claim 86. The motivation for combining the references presented in claim 86 applies to this claim and is incorporated herein by reference.

13. Claims 113, 114 and 119 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,714,201 to Grinstein et al. in view of U.S. Patent No. 6,414,685 to Takakura et al. in view of U.S. Patent No. 6,690,376 to Saito et al.

14. Regarding claim 113, Grinstein et al. disclose "in a computer-implemented animation system, a method for animating an object, (*Fig. 3*) the method comprising:

- r. receiving an input (*receiving the instructions written in C++ that invoke the OpenMotion API as described in section 6.2 at column 15*) specifying a Grow/Shrink behavior (*motion behaviors are described in section 6.2.6, where it is disclosed "A Behavior is an action that changes a motion's parameters" in lines 12-14 of column 29*),
- s. the Grow/Shrink behavior indicating how to change a value of a scale parameter of the object over time (*scale parameter is defined in Table 10 in column 26, as is the growth parameter, "the time derivative of scale, ds/dt ," both of which can be controlled by the Behavior according to the interactive controller actions `scaleControl`, `growthControl`, and `growthRateControl` as described in Table 14 of column 29, and described in 6.2.6.1 in column 29*) by either
 - xx. changing a size of the object by a steady amount per second (*interactive controller `growthControl` continually adapts the object to an increase in size according to the input parameter, as described in lines 30-34 of column 29, where the input parameter is a vector $[s_x, s_y, s_z]$ which specifies the time derivative of scale ds/dt in each direction x, y, z ; time in the OpenMotion API is specified per second by default as shown in Table 4 of column 17, and further shown in line 13 of column 26, and the bottom of column 36*) or
 - xxi. changing the object's size from an original size to a final size (*interactive controller `scaleControl` continually adapts the object from an original size to the input parameter, as described in lines 30-34 of column 29, where the input parameter is a vector $[s_x, s_y, s_z]$ which establishes a final size*);

t. animating the object by changing the value of the scale parameter of the object over time according to the Grow/Shrink behavior (*lines 46-53 of column 75:*

"...generating an animated view of the given model in which the given model is rendered at each of a succession of time values with individual ones of the model's nodes being shown in each successive rendering as having a position and orientation determined as a function of the value for the rendering's corresponding time value of the position and orientation values defined by the node's associated motion..."; section 6.2.6 describes Behaviors which changing the value of a parameter, for example scale/growth, over time); and

u. outputting the animated object (*the end result of applying motions and behaviors is an animated image of an object as shown in Fig. 3 and described in lines 17-20 of column 53: "Since Mojo is a real-time motion editor, the model of the running man is shown moving according to a set of motions that have been applied to the individual nodes of its hierarchical model."*);

v. wherein the Grow/Shrink behavior can be configured regarding

xxii. if the object's size is changing by a steady *amount* per second, a first number and a second number, wherein the first number is *a size* added to the object's horizontal size per second, and wherein the second number is *a size* added to the object's vertical size each second (*growthControl controller parameter takes an as input a vector $[s_x, s_y, s_z]$, components corresponding to the first and second number, the vector specifies the time derivative of scale ds/dt in*

each direction x, y, z; time in the OpenMotion API is specified per second as shown in Table 4 of column 17);

xxiii. an acceleration with which the object's size changes over time
(growthRateControl parameter in Table 14 controls the growth rate described as the time derivative of growth dg/dt in Table 10 in column 26).

15. Regarding the first parameter, Grinstein et al. does not disclose the size is a number of pixels.
16. Saito specifies that objects are defined by pixel data (*Fig. 4*) with a defined height and width in terms of pixels (*lines 20-23 of column 11: "...the cell size indicating the number of pixels in vertical and horizontal directions 205;...."*).
17. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to represent Grinstein's objects using Saito's sprites with dimensions specified as pixels, and thus, obtaining a parameter that can be configured regarding "if the object's size is changing by a steady number of pixels per second, a first number of horizontal pixels and a second number of vertical pixels, wherein the first number is a size added to the object's horizontal size per second, and wherein the second number is a size added to the object's vertical size each second." The motivation for doing so would have been to eliminate for commands to render 3D objects to 2D images, which would in turn, lessen the computational resources needed to create the animation.
18. Regarding the second parameter, Grinstein et al. does not disclose "if the object's size is changing from the original size to the final size, a first number of horizontal pixels and a second

number of vertical pixels, wherein the first number represents horizontal pixels in the final size, and where the second number represents vertical pixels in the final size.”

19. Takakura et al. disclose if the object's size is changing from the original size (*634 in Fig. 43A*) to the final size (*636 in Fig. 43B*), configuring a set of coordinates, wherein the set of coordinates represents in the final size (*664 shows the final coordinates of the object 636, which 634 will be scaled to*). Takahura et al. represent objects using coordinate data and not pixels. Saito discloses pixel data.

20. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to further apply Takakura's teachings of setting the final size to Saito's sprites in Grinstein's system, which would arrive at the limitation ““if the object's size is changing from the original size to the final size, a first number of horizontal pixels and a second number of vertical pixels, wherein the first number represents horizontal pixels in the final size, and where the second number represents vertical pixels in the final size.” The motivation for doing so would have been to allow the user to intuitively set the size of Saito's sprite, because Saito sprite is defined in terms of the number of pixels in the horizontal and vertical directions, and thus, it would have been natural to define a change in size in terms of the same units that the object is defined, just as Takakura defines a change in size in terms of the coordinates. Therefore, it would have been obvious to modify Grinstein's behavior with Saito's sprites and Takakura's method of scaling graphical objects to obtain the invention specified in claim 113.

21. Claim 119 recites limitations similar in scope to the limitations of claim 113, but as a method on a computer program product. As shown in the rejection, the combination of Grinstein et al., Takakura et al., and Saito et al. disclose the limitations of claim 86. Additionally, Grinstein

et al. discloses a computer program product for animating an object, the computer program product comprising a computer-readable storage medium containing computer program code in lines 51-56 of column 6: "It should be understood that the invention is meant to include apparatus, methods, computer programming recorded on computer readable media, and propagated signals capable of providing functionality of the type recited in each claim, even if there currently are not claim covering each of these different class of inventions below." Thus, the computer readable medium recited in claim 119 is met by the combination of Grinstein et al., Takakura et al., and Saito et al. according to the mapping presented in the rejection of claim 113 because the computer readable medium stores the method disclosed in claim 113. The motivation for combining the references presented in claim 113 applies to this claim and is incorporated herein by reference.

22. Regarding claim 114, Grinstein et al. further disclose "a number of frames by which to offset an end of the Grow/Shrink behavior's effect relative to a last frame of a position of the Grow/Shrink behavior (*Table 9 of column 25 shows an "offset" variable part of the cyclic temporal predicate; section 6.2.2.2.2 section "Simulation clock" describes how the frames of animation are generated in relation to the application time*) in a timeline" (*lines 34-39 of column 17 describe a timeline in established by the Clock*). The motivation for combining the references presented in the parent claim applies to this claim and is incorporated herein by reference.

Continued Examination Under 37 CFR 1.114

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e)

has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 1/19/2009 has been entered.

Response to Arguments

5. Applicant's arguments with respect to the pending have been considered but are moot in view of the new rationale presented in this Office Action, which was necessitated by the amendment.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason M. Repko whose telephone number is (571)272-8624. The examiner can normally be reached on Monday through Friday 8:00 am - 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Xiao M. Wu can be reached on (571)272-7761. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 10/826,973
Art Unit: 2628

Page 22

/Jason M Repko/
Examiner, Art Unit 2628